SAT vs CSP: a commentary*

Toby Walsh

TU Berlin and UNSW Sydney
Germany and Australia
tw@cse.unsw.edu.au

Abstract. In 2000, I published a relatively comprehensive study of mappings between propositional satisfiability (SAT) and constraint satisfaction problems (CSPs) [Wal00]. I analysed four different mappings of SAT problems into CSPs, and two of CSPs into SAT problems. For each mapping, I compared the impact of achieving arc-consistency on the CSP with unit propagation on the corresponding SAT problems, and lifted these results to CSP algorithms that maintain (some level of) arc-consistency during search like FC and MAC, and to the Davis-Putnam procedure (which performs unit propagation at each search node). These results helped provide some insight into the relationship between propositional satisfiability and constraint satisfaction that set the scene for an important and valuable body of work that followed. I discuss here what prompted the paper, and what followed.

1 Introduction

Two decades after it was written, a paper I wrote comparing propositional satisfiability and constraint satisfaction [Wal00] has, to my surprise, become one of the most cited works from the CP 2000 conference (with 353 cites currently on Google Scholar). Indeed, to my surprise, it currently ranks second in citations of *all* of my scientific papers. In part, this is luck. There were a number of encodings of CSPs into SAT that were well known but hadn't been documented extensively in prior work. It was also luck that had me write these down as I nearly didn't submit the paper since I remember thinking that many of its results were not very deep. And it was luck that I had started working in the constraint programming community as a result of having come from the area of propositional satisfiability. This put me in the natural place to draw the connections between the two areas. It was obvious to me that arc consistency and unit propagation were making very similar inferences. But perhaps this seemed less immediate at that time to people without feet in both areas. This paper attempted to correct this.

2 Contributions

This wasn't the first paper to look at mappings between SAT and CSPs. For instance, Bennaceur had previously looked at encoding SAT problems as CSPs [Ben96], whilst Génisson and Jégou had looked at encoding CSPs as SAT problems [GJ96]. However,

* The original CP 2000 paper is available without pay-wall at https://tinyurl.com/SATvCSP

earlier studies like these has only considered a limited number of mappings. This was perhaps the first study to attempt to look more comprehensively about mappings between the two domains, and certainly to do so in both directions at once. The paper considered 4 encodings of SAT into CSPs (dual, hidden variable, literal and the non-binary encoding), 2 encodings of CSPs into SAT (direct and log encoding), and studied some of the theoretical properties of these mappings.

To be more specific, I proved that achieving arc-consistency on the dual encoding does more work than unit propagation on the original SAT problem, whilst achieving arc-consistency on the hidden variable and literal encodings does essentially the same work. I then extended these results to algorithms that maintain some level of arc-consistency during search like forward checking (FC) and maintaining arc-consistency (MAC), and DP which performs unit propagation at each search node supposing equivalent branching heuristics. I proved that the Davis Putnam (DP) algorithm strictly dominates FC applied to the dual encoding, is incomparable to MAC applied to the dual encoding, explores the same number of branches as MAC applied to the hidden variable encoding, and strictly dominates MAC applied to the literal encoding. Finally I proved that unit propagation on the direct encoding does less work than achieving arc-consistency on the original problem, but more work than unit propagation on the log encoding. DP on the direct encoding explores the same size search tree as FC applied to the original problem, but is strictly dominated by MAC. By comparison, DP on the log encoding is strictly dominated by both FC and MAC applied to the original problem. A follow up paper by my colleague, Ian Gent looked at the closely related support encoding of CSPs into SAT [Gen02]. As the name suggest, this encodes not conflicts (as the dual and hidden-variable encodings do), but supports.

3 What happened next?

Though these results were not deep, they did set the scene for a range of future research that brought ideas and techniques from propositional satisfiability into constraint satisfaction. Interestingly, the direction has been almost all one way. I am not aware of much work exporting ideas and techniques from constraint satisfaction into propositional satisfiability. I leave it as an interesting open question for the reader as to why this might be the case.

3.1 Solving CSPs by encoding into SAT

The most immediate and direct application of the encodings discussed in [Wal00] and [Gen02] is to encode CSPs into SAT so we can bring to bear the advanced and powerful SAT solvers based on complete and local search methods that are now available to download. For instance, we showed that it was effective to encode global GRAMMAR and REGULAR constraints into SAT [QW07], and gave state of the art results on some shift rostering problems. Similar encodings of CSPs into answer set programs (ASPs) [DW10,DW11b,DW11a] and into integer linear programs (ILPs) [HKW03] have also been shown to be highly effective. Many medal winners in the MiniZinc constraint solving competition have used SAT encodings (e.g. PicatSAT) or SAT inspired algorithms (e.g. HaifaCSP and MinisatID).

3.2 Lazy clause generation

Perhaps the most important work that followed is the research of Ohrimenko, Stuckey, Codish and colleagues on lazy clause generation [OSC09]. This shows that domain propagation on a CSP can be achieved by generating clauses for a SAT solver. Since a naive static translation is impractical except in limited cases, the clauses are generated lazily as needed. Such solving techniques open up many new possibilities like nogood learning and the use of SAT based branching heuristics. Lazy clause solvers remain today some of the fastest on many classes of challenging problems.

3.3 Analysing global constraints

The final area I want to mention is one that hasn't received the attention I believe it deserves since it touches on one of the aspects of constraint solving that makes it special: namely, global constraints. Encodings of constraints into SAT have played a central role in understanding how far we can effectively decompose global constraints. There are a number of promising results decomposing global constraints into smaller, and easier to propagate constraints (e.g. [GSW00a,GSW00b,QW07,BNQ$^+$07,BKN$^+$09,BKNW10]). However, by drawing on lower bound results from circuit complexity, we proved some fundamental limits to such decompositions [BKNW09].

Consider encodings of global constraints into propositional formulae in conjunctive normal form that have the fundamental property that unit propagation on the decomposition enforces the same level of consistency as a specialized propagation algorithm like that which maintains arc consistency. We proved that a constraint propagator has a a polynomial size decomposition if and only if it can be computed by a polynomial size monotone Boolean circuit. Lower bounds on the size of monotone Boolean circuits thus translate to lower bounds on the size of decompositions of global constraints. For instance, we proved that there is no polynomial sized decomposition of the domain consistency propagator for the ALLDIFFERENT constraint. On the other hand, bound and range consistency propagators of ALLDIFFERENT can be decomposed effectively into SAT, as we demonstrate in [BKN$^+$09].

4 Conclusions

Constraint satisfaction and propositional satisfiability are closely related areas. Mappings between the two have proved a fruitful area for research in the last two decades. Somewhat surprisingly, it has mostly been an one-way flow, with tools and techniques from propositional satisfiability being used to help solve constraint satisfaction problems. It's interesting to consider if we could not also look at the reverse direction?

Acknowledgments

This research was funded by the European Research Council under the Horizon 2020 Programme via AMPLify 670077.

References

[Ben96] H. Bennaceur. The satisfiability problem regarded as a constraint satisfaction problem. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 155–159. European Conference on Artificial Intelligence, Wiley, 1996.

[BKN+09] C. Bessiere, G. Katsirelos, N. Narodytska, C-G. Quimper, and T. Walsh. Decompositions of all different, global cardinality and related constraints. In *Proceedings of 21st IJCAI*, pages 419–424. International Joint Conference on Artificial Intelligence, 2009.

[BKNW09] C. Bessiere, G. Katsirelos, N. Narodytska, and T. Walsh. Circuit complexity and decompositions of global constraints. In *Proceedings of 21st IJCAI*, pages 412–418. International Joint Conference on Artificial Intelligence, 2009.

[BKNW10] C. Bessiere, G. Katsirelos, N. Narodytska, and C-G. Quimper T. Walsh. Decomposition of the NValue constraint. In D. Cohen, editor, *Proceedings of the 16th International Conference on the Principles and Practice of Constraint Programming (CP 2010)*, volume 6308 of *Lecture Notes in Computer Science*. Springer, 2010.

[BNQ+07] S. Brand, N. Narodytska, C.-G. Quimper, P. Stuckey, and T. Walsh. Encodings of the sequence constraint. In *13th International Conference on Principles and Practices of Constraint Programming (CP-2007)*. Springer-Verlag, 2007.

[DW10] C. Drescher and T. Walsh. A translational approach to constraint answer set solving. *Theory and Practice of Logic Programming*, 10(4-6):465–480, 2010.

[DW11a] C. Drescher and T. Walsh. Modelling GRAMMAR constraints with answer set programming. In *Proceedings of the 27th International Conference on Logic Programming (ICLP 2011)*, 2011.

[DW11b] C. Drescher and T. Walsh. Translation-based constraint answer set solving. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*. International Joint Conference on Artificial Intelligence, 2011.

[Gen02] I.P. Gent. Arc consistency in SAT. In F. van Harmelen, editor, *Proceedings of ECAI-2002*. IOS Press, 2002.

[GJ96] R. Genisson and P. Jegou. Davis and Putnam were already forward checking. In W. Wahlster, editor, *Proceedings of the 12th ECAI*, pages 180–184. European Conference on Artificial Intelligence, Wiley, 1996.

[GSW00a] I.P. Gent, K. Stergiou, and T. Walsh. Decomposable constraints. *Artificial Intelligence*, 123(1-2):133–156, 2000.

[GSW00b] I.P. Gent, K. Stergiou, and T. Walsh. Decomposable constraints. In K. Apt, A. Kakis, E. Monfroy, and F. Rossi, editors, *New Trends in Constraints*, number 1865 in LNAI, pages 134–149. Springer, 2000.

[HKW03] B. Hnich, Z. Kiziltan, and T. Walsh. Hybrid modelling for robust solving. *Anals of Operations Research*, 130(1-4):19–39, 2003.

[OSC09] Olga Ohrimenko, Peter J. Stuckey, and Michael Codish. Propagation via lazy clause generation. *Constraints*, 14(3):357–391, 2009.

[QW07] C.-G. Quimper and T. Walsh. Decomposing global grammar constraints. In *13th International Conference on Principles and Practices of Constraint Programming (CP-2007)*. Springer-Verlag, 2007.

[Wal00] T. Walsh. SAT v CSP. In Rina Dechter, editor, *6th International Conference on Principles and Practices of Constraint Programming (CP-2000)*, pages 441–456. Springer-Verlag, 2000.